

AZ-104T00A

Module 09:

Serverless Computing



Module Overview

- Lesson 01: Azure App Service Plans
- Lesson 02: Azure App Services
- Lesson 03: Container Services
- Lesson 04: Azure Kubernetes Service
- Lesson 05: Module 09 Lab and Review

Lesson 01: Azure App Service Plans



Azure App Service Overview

- Azure App Service Plans
- App Service Plan Pricing Tiers
- App Service Plan Scaling
- App Service Plan Scale Out
- Demonstration - Create an App Service Plan

Azure App Service Plans

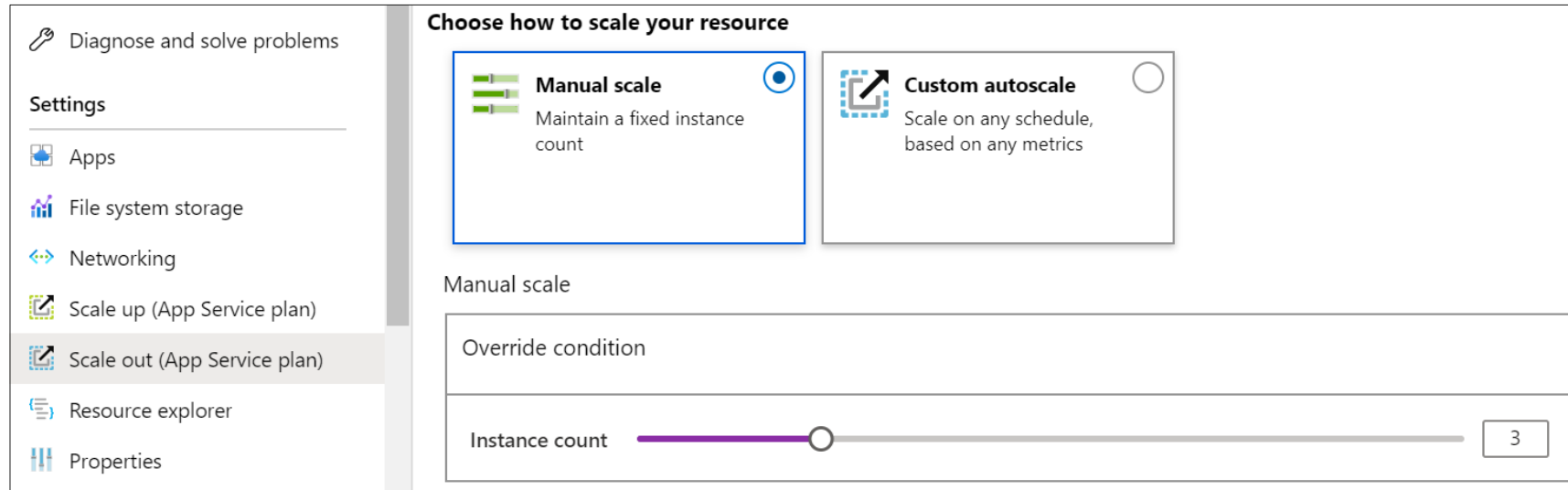
- Define a set of compute resources for a web app to run
- Determines performance, price, and features
- One or more apps can be configured to run in the same App Service plan
- App Service plans define:
 - Region where compute resources will be created
 - Number of virtual machine instances
 - Size of virtual machine instances (Small, Medium, Large)
 - Pricing tier (next slide)

App Service Plan Pricing Tiers

Selected Features	Free	Shared (dev/test)	Basic (dedicated dev/test)	Standard (production workloads)	Premium (enhanced scale and performance)	Isolated (high-performance, security and isolation)
Web, mobile, or API apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Auto Scale	–	–	–	Supported	Supported	Supported
Deployment Slots	0	0	0	5	20	20
Max Instances	-	-	Up to 3	Up to 10	Up to 30	Up to 100

- **Shared compute** (Free and Shared). Run apps on the same Azure VM as other App Service apps, and the resources cannot scale out
- **Dedicated compute** (Basic, Standard, Premium). Run apps in the same plan in dedicated Azure VMs
- **Isolated**. Runs apps on dedicated Azure VMs in dedicated Azure virtual networks

App Service Plan Scaling



- Scale up (change the App Service plan)
 - More hardware (CPU, memory, disk)
 - More features (dedicated virtual machines, staging slots, autoscaling)
- Scale out (increase the number of VM instances)
 - Manual (fixed number of instances)
 - Autoscale (based on predefined rules and schedules)

App Service Plan Scale Out

Default Auto created scale condition 

Delete warning  The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode ☒ Scale based on a metric ☐ Scale to a specific instance count

Rules

 No metric rules defined; click hyperlink [Add a rule](#) to scale out and scale in your instances based on rules. For example: 'Add a rule that increases instance count by 1 when CPU percentage is above 70%'.
[+ Add a rule](#)

Instance limits

Minimum  Maximum  Default 

Schedule **This scale condition is executed when none of the other scale condition(s) match**

- Adjust available resources based on the current demand
- Improves availability and fault tolerance
- Scale based on a metric (CPU percentage, memory percentage, HTTP requests)
- Scale according to a schedule (weekdays, weekends, times, holidays)
- Can implement multiple rules – combine metrics and schedules
- Don't forget to scale down

Demonstration – Create an App Service Plan

- Create an App Service Plan in the Azure Portal
- Review Pricing Tiers
- Configure Autoscaling

Lesson 02: Azure App Services



Managing App Services Overview

- Azure App Service
- Creating an App Service
- Continuous Deployment
- Deployment Slots
- Creating Deployment Slots
- Securing an App Service
- Custom Domain Names
- Backup an App Service
- Application Insights
- Demonstration – Create an App Service

Azure App Service



.NET



Node.js



PHP



Java



Python (on Linux)



HTML



Custom Windows container (Preview)

- Includes Web Apps API Apps, Mobile Apps, and Function apps
- Fully managed environment enabling high productivity development
- Platform-as-a-service (PaaS) offering for building and deploying highly available cloud apps for web and mobile
- Platform handles infrastructure so developers focus on core web apps and services
- Developer productivity using .NET, .NET Core, Java, Python and a host of others
- Provides enterprise-grade security and compliance

Creating an App Service

- Name must be unique
- Access using *azurewebsites.net* – can map to a custom domain
- Publish Code (Runtime Stack)
- Publish Docker Image (Image source)
- Linux or Windows
- Region closest to your users
- App Service Plan

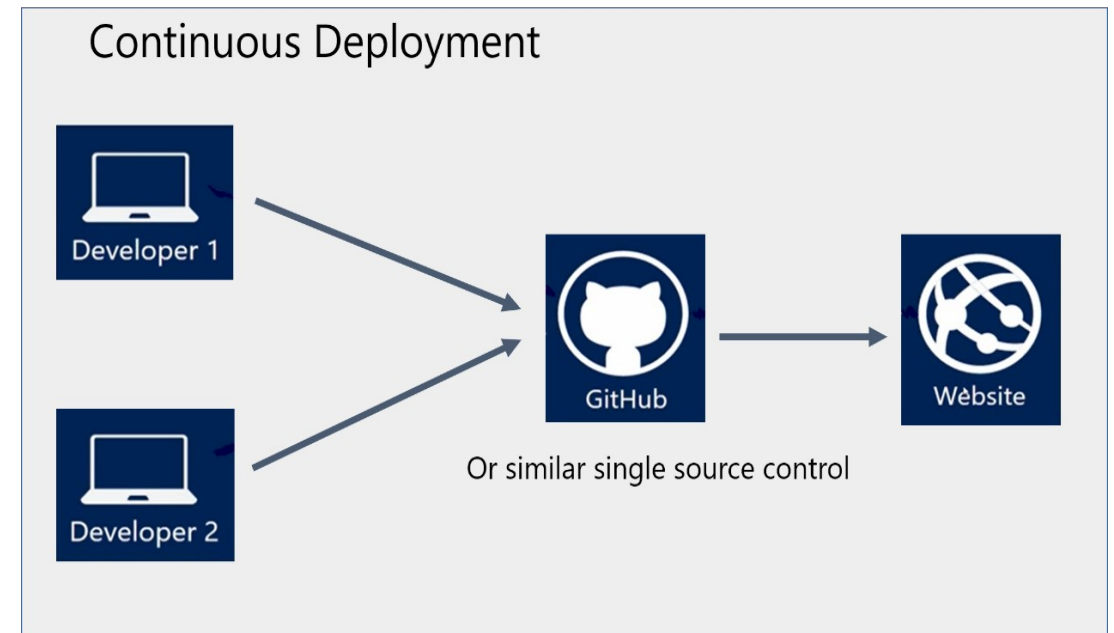
Web App * Basics Tags Review and create

Project Details
* Subscription ⓘ Concierge Subscription
* Resource Group ⓘ Learn-11111111-2222-3333-4444-555555555555
[Create new](#)

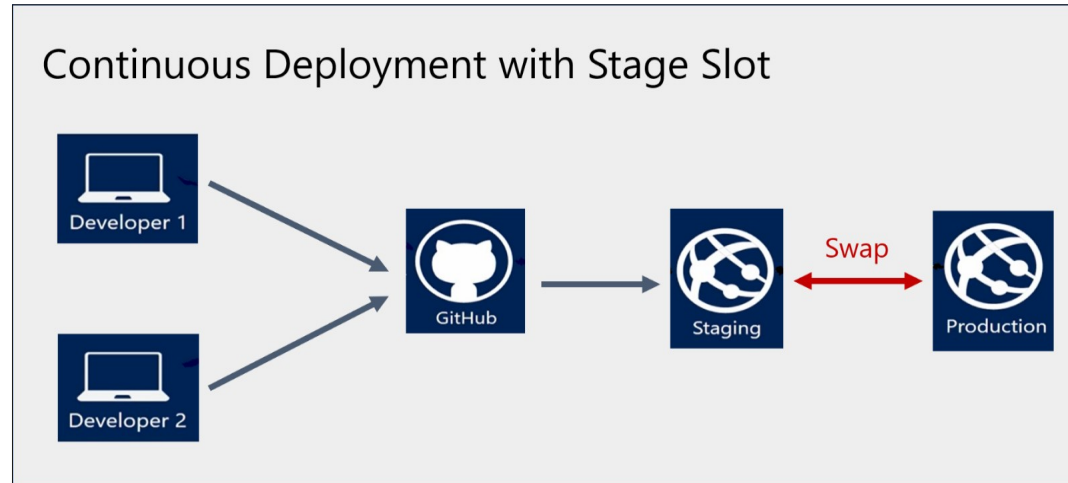
Instance Details
* Name your-app-name ✓
.azurewebsites.net
* Publish Code Docker Image
* Runtime stack .NET Core 2.2
* Operating System Linux Windows
* Region Central US
App Service Plan
* Linux Plan (Central US) ⓘ (New) ASP-Learn-11111111-2222-3333-4444
[Create new](#)
* Sku and size
Free F1
1 GB memory
[Change size](#)

Continuous Deployment

- Work in a single source control
- Whenever code updates are pushed to the source control, then the website or web app will automatically pick up the updates
- A continuous deployment workflow publishes the most recent updates from a project
- Use the portal for continuous deployments from GitHub, Bitbucket, or Visual Studio Team Services



Deployment Slots

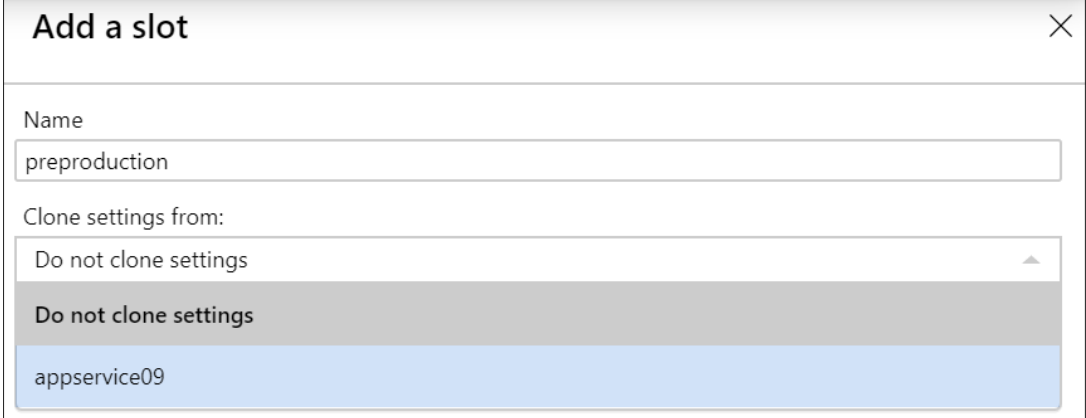


Service Plan	Slots
Free, Shared, Basic	0
Standard	Up to 5
Premium	Up to 20
Isolated	Up to 20

- Deploy to a different deployment slots (depends on service plan)
- Validate changes before sending to production
- Deployment slots are live apps with their own hostnames
- Avoids a cold start – eliminates downtime
- Fallback to a last known good site
- Auto Swap when pre-swap validation is not needed

Creating Deployment Slots

- A new slot can be empty or cloned
- When you clone, pay attention to the settings
 - Slot-specific app settings and connection strings
 - Continuous deployment settings
 - App Service authentication settings
- Not all settings are sticky (endpoints, custom domain names, SSL certificates, scaling)
- Review and edit your settings before swapping



The screenshot shows a dialog box titled "Add a slot" with a close button (X) in the top right corner. Inside the dialog, there is a "Name" label followed by a text input field containing the text "preproduction". Below this, there is a "Clone settings from:" label followed by a dropdown menu. The dropdown menu is open, showing three options: "Do not clone settings" (which is currently selected and highlighted in grey), "Do not clone settings" (in a darker grey), and "appservice09" (highlighted in blue).

Securing an App Service

- **Authentication**

- Enable authentication – default anonymous
- Log in with a third-party identity provider

- **Security**

- Troubleshoot with Diagnostic Logs – failed requests, app logging
- Add an SSL certificate – HTTPS
- Define a priority ordered allow/deny list to control network access to the app
- Store secrets in the Azure Key Vault

App Service Authentication

☐ Off ☒ On

Action to take when request is not authenticated

Log in with Azure Active Directory

Allow Anonymous requests (no action)

Log in with Azure Active Directory

Log in with Microsoft Account

Log in with Facebook

Log in with Google

Log in with Twitter

 **Protocol Settings**

Protocol settings are global and apply to all bindings defined by your app.

HTTPS Only: ☐ Off ☒ On

Minimum TLS Version: ☐ 1.0 ☐ 1.1 ☒ 1.2

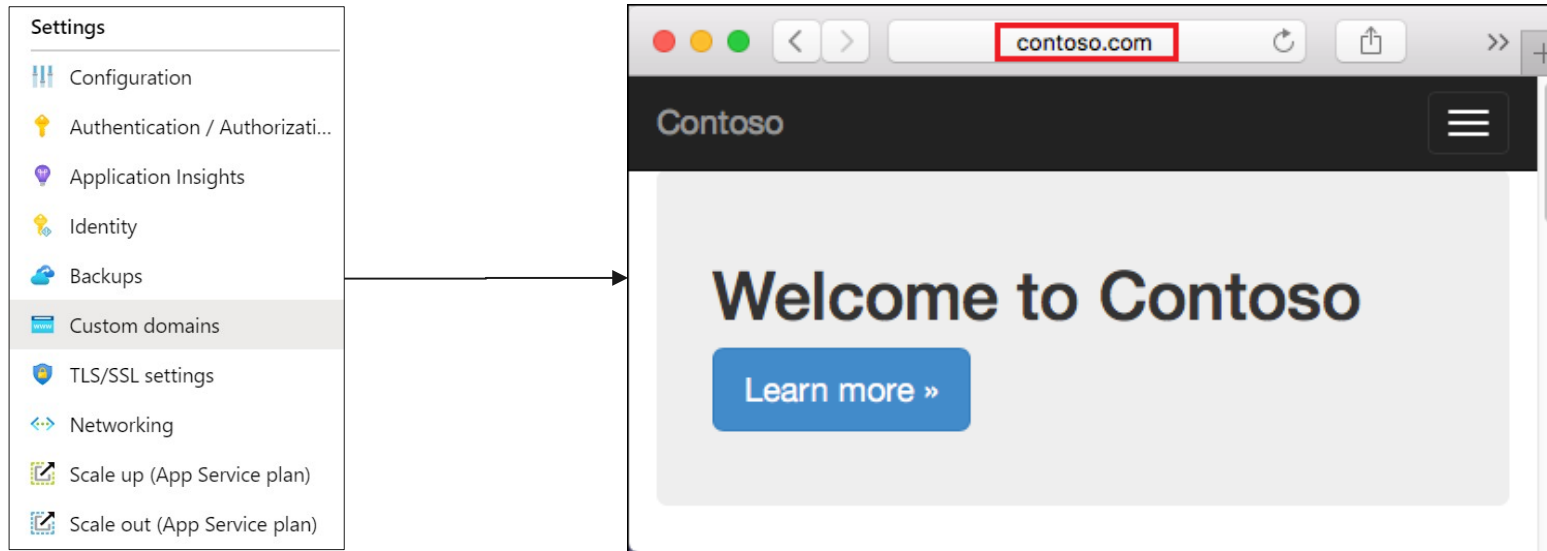
 **TLS/SSL bindings**

Bindings let you specify which certificate to use when responding to requests to a specific hostname over HTTPS. TLS/SSL Binding requires valid private certificate (.pfx) issued for the specific hostname. [Learn more](#)

+ Add TLS/SSL Binding

☐ Host name	Private Certificate Thumbp...	TLS/SSL Type
No TLS/SSL bindings configured for the app.		

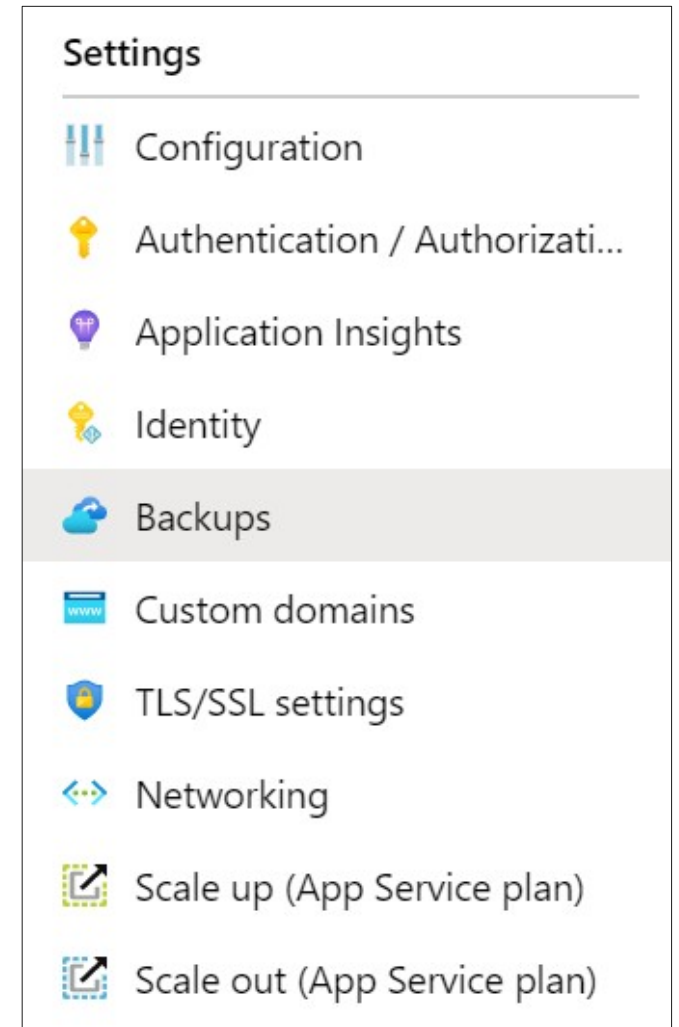
Custom Domain Names



- Redirect the default web app URL
- Validate the custom domain in Azure
- Use the DNS registry for your domain provider – create a CNAME or A record with the mapping
- Ensure App Service plan supports custom domains

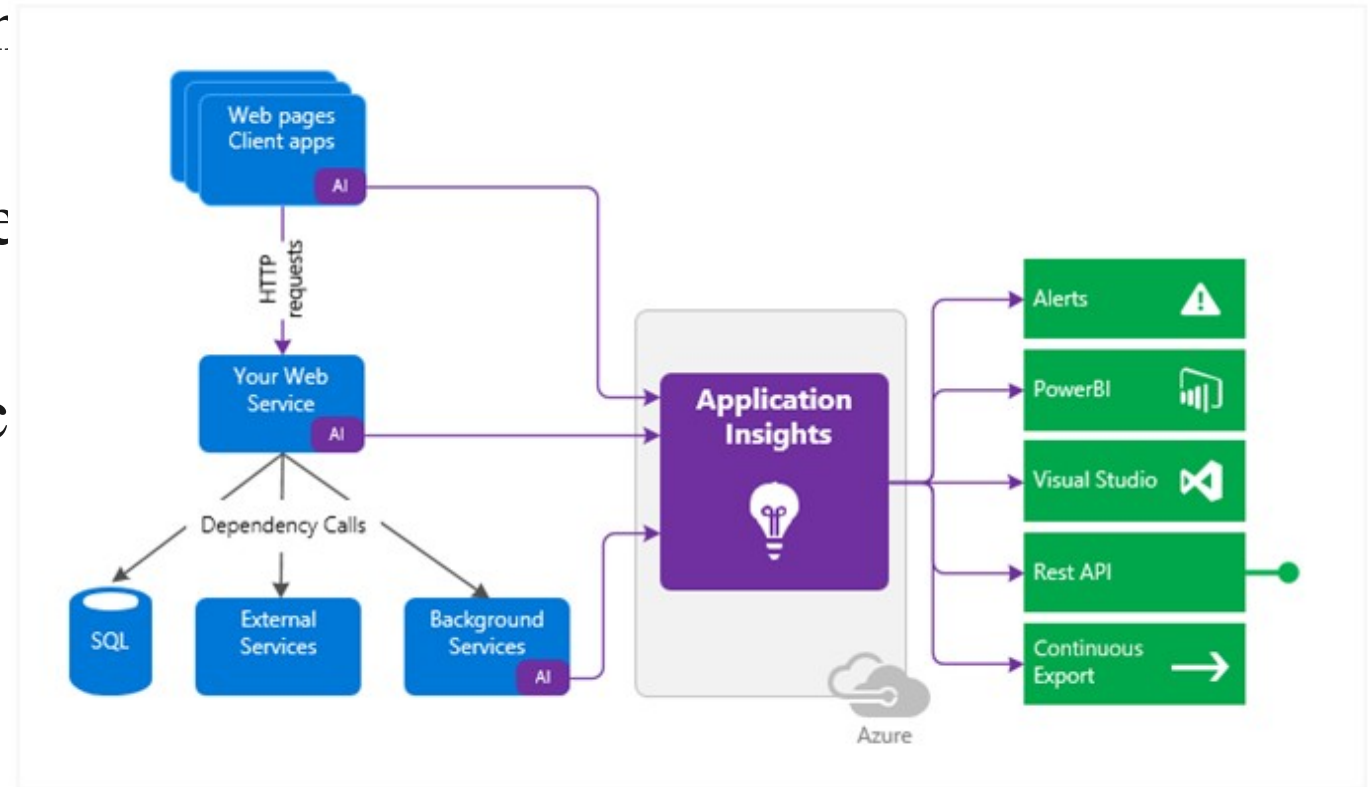
Backup an App Service

- Create app backups manually or on a schedule
- Backup the configuration, file content, and database connected to the app
- Requires Standard or Premium plan
- Backups can be up to 10 GB of app and database content
- Configure partial backups and exclude items from the backup
- Restore your app on-demand to a previous state, or create a new app



Application Insights

- Request rates, response times, and failure rates
- Dependency rates, response time and failure rates
- Page views and load performance
- User and session counts
- Performance counters
- Diagnostics and Exceptions



Demonstration – Create an App Service

- Create a Web App in the Azure Portal
- Test the Web App
- Configure Deployment Slots
- Configure Backup

Lesson 03: Container Services



Container Services Overview

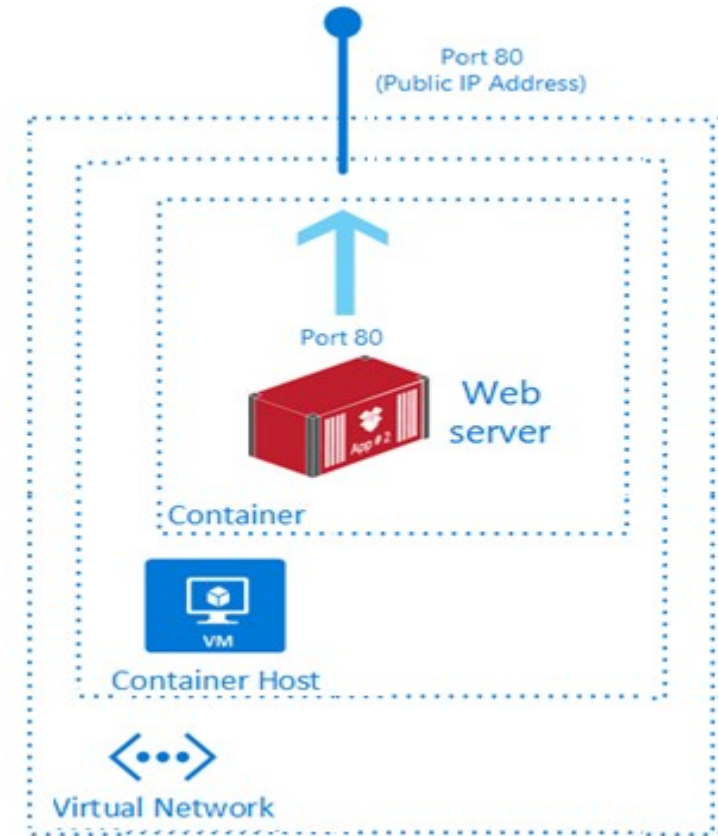
- Containers vs. Virtual Machines
- Azure Container Instances
- Container Groups
- Docker

Containers vs Virtual Machines

Feature	Containers	Virtual Machines
Isolation	Typically provides lightweight isolation from the host and other containers but doesn't provide as strong a security boundary as a virtual machine.	Provides complete isolation from the host operating system and other VMs. This is useful when a strong security boundary is critical, such as hosting apps from competing companies on the same server or cluster.
Operating system	Runs the user mode portion of an operating system and can be tailored to contain just the needed services for your app, using fewer system resources.	Runs a complete operating system including the kernel, thus requiring more system resources (CPU, memory, and storage).
Deployment	Deploy individual containers by using Docker via command line; deploy multiple containers by using an orchestrator such as Azure Kubernetes Service.	Deploy individual VMs by using Windows Admin Center or Hyper-V Manager; deploy multiple VMs by using PowerShell or System Center Virtual Machine Manager.
Persistent storage	Use Azure Disks for local storage for a single node, or Azure Files (SMB shares) for storage shared by multiple nodes or servers.	Use a virtual hard disk (VHD) for local storage for a single VM, or an SMB file share for storage shared by multiple server.
Fault tolerance	If a cluster node fails, any containers running on it are rapidly recreated by the orchestrator on another cluster node.	VMs can fail over to another server in a cluster, with the VM's operating system restarting on the new server.

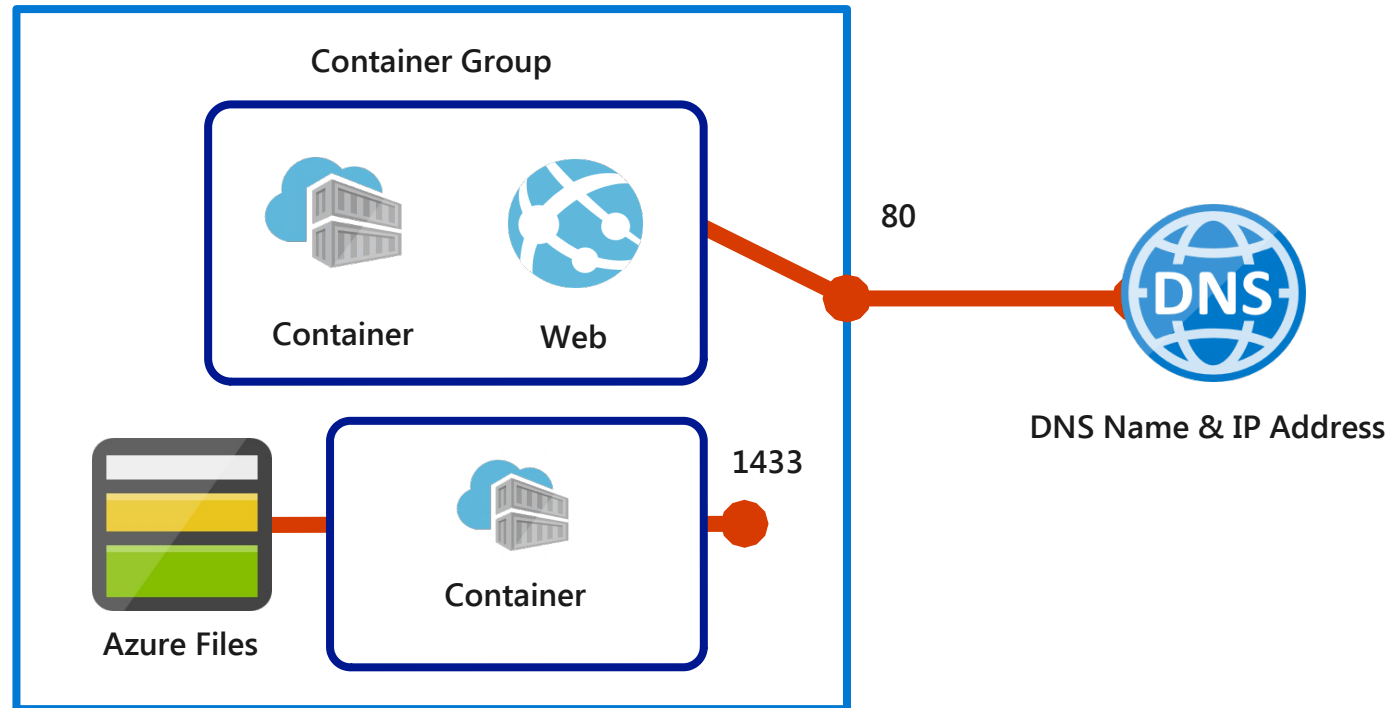
Azure Container Instances

- PaaS Service
- Fast startup times
- Public IP connectivity and DNS name
- Hypervisor-level security
- Isolation features
- Custom sizes
- Persistent storage
- Linux and Windows Containers
- Co-scheduled Groups
- Virtual network Deployment



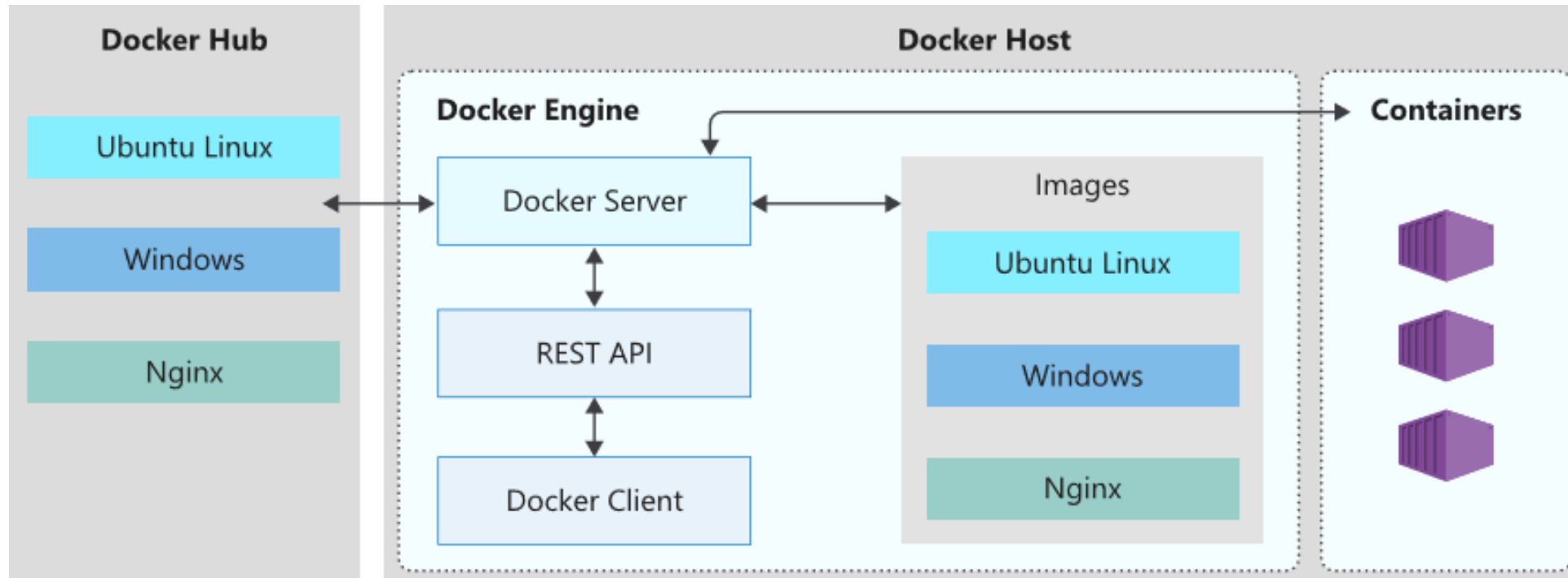
Fastest way to run a container in Azure
without provisioning a VM

Container Groups



- Top-level resource in Azure Container Instances
- A collection of containers that get scheduled on the same host
- The containers in the group share a lifecycle, resources, local network, and storage volumes

Docker



- Enables developers to host applications within a container
- A container is a standardized "unit of software" that contains everything required for an application to run
- Available on both Linux and Windows and can be hosted on Azure

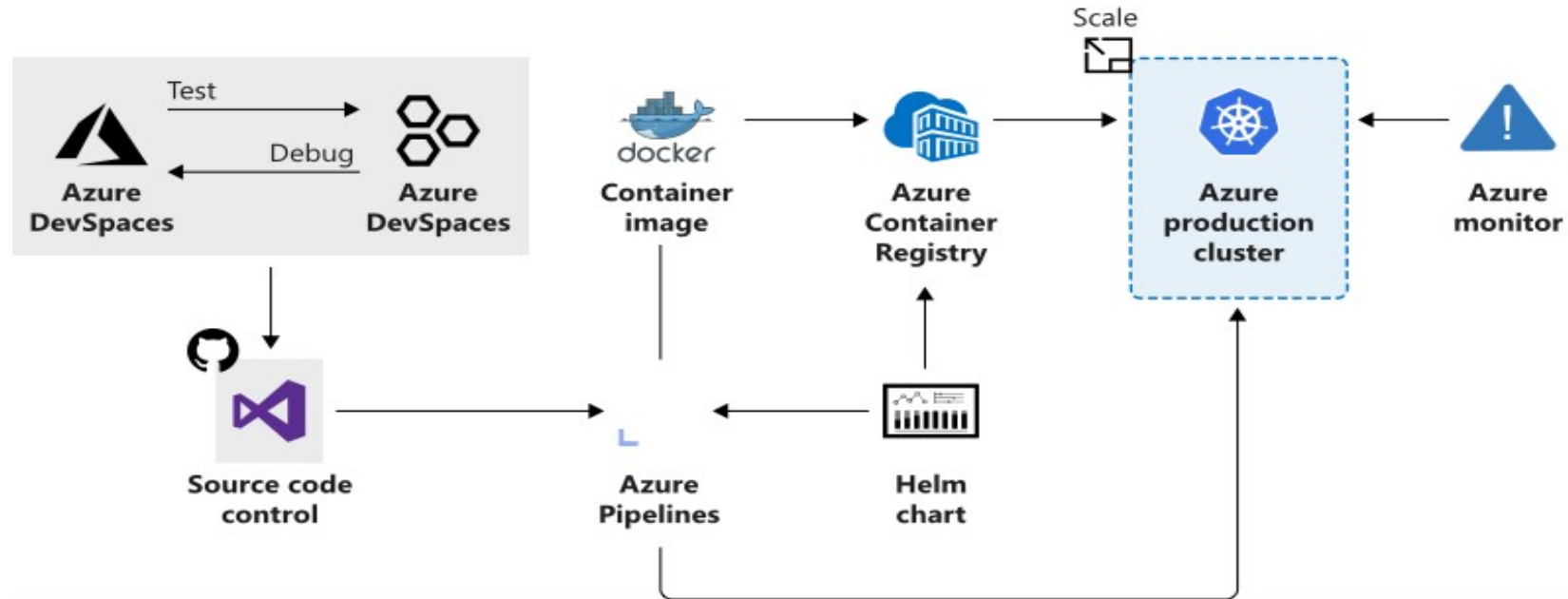
Lesson 04: Azure Kubernetes Service



Azure Kubernetes Services Overview

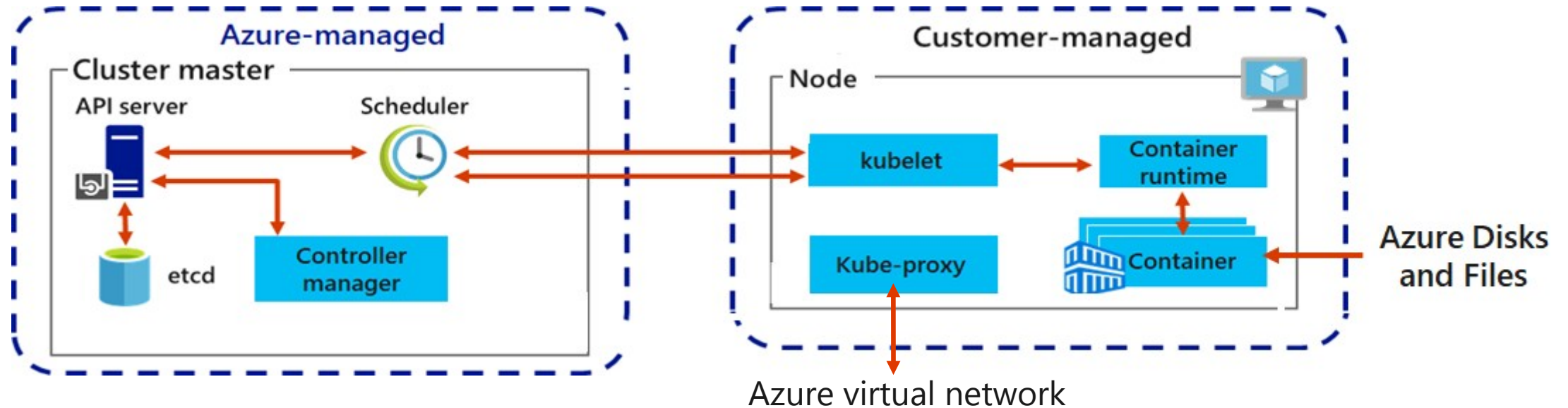
- Azure Kubernetes Services
- AKS Clusters and Nodes
- AKS Networking
- AKS Storage
- AKS Security
- AKS and Azure Active Directory
- AKS Scaling
- AKS Scaling to ACI
- Virtual Kubelet
- Demonstration – Deploy Azure Kubernetes Service

Azure Kubernetes Service



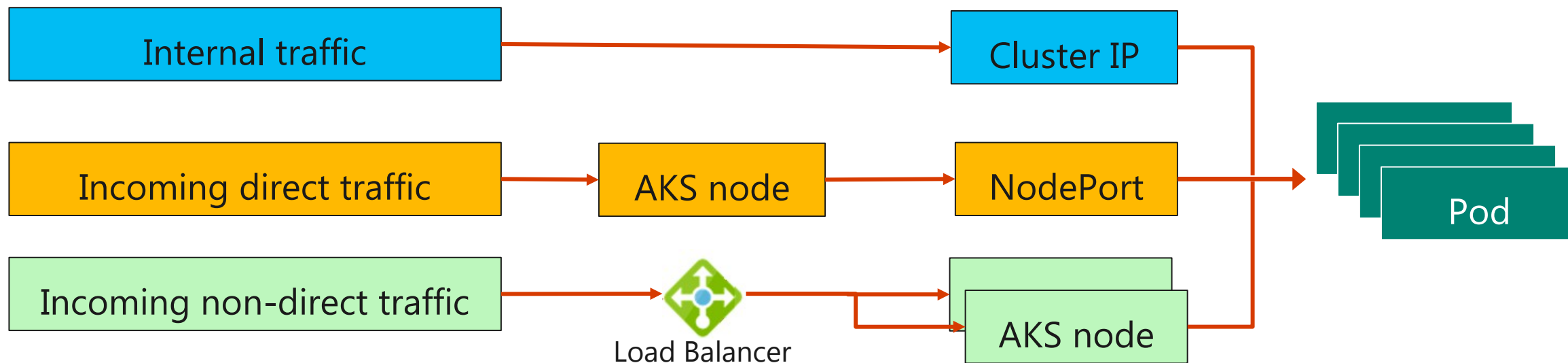
- Manages health monitoring and maintenance
- Performs simple cluster scaling
- Enables master nodes to be fully managed by Microsoft
- You're responsible only for managing the agent nodes
- Master nodes are free, and you pay only for running agent nodes

AKS Clusters and Nodes



- Cluster master provides core Kubernetes services and orchestration
- Nodes run applications and supporting services
- Each individual node is an Azure virtual machine

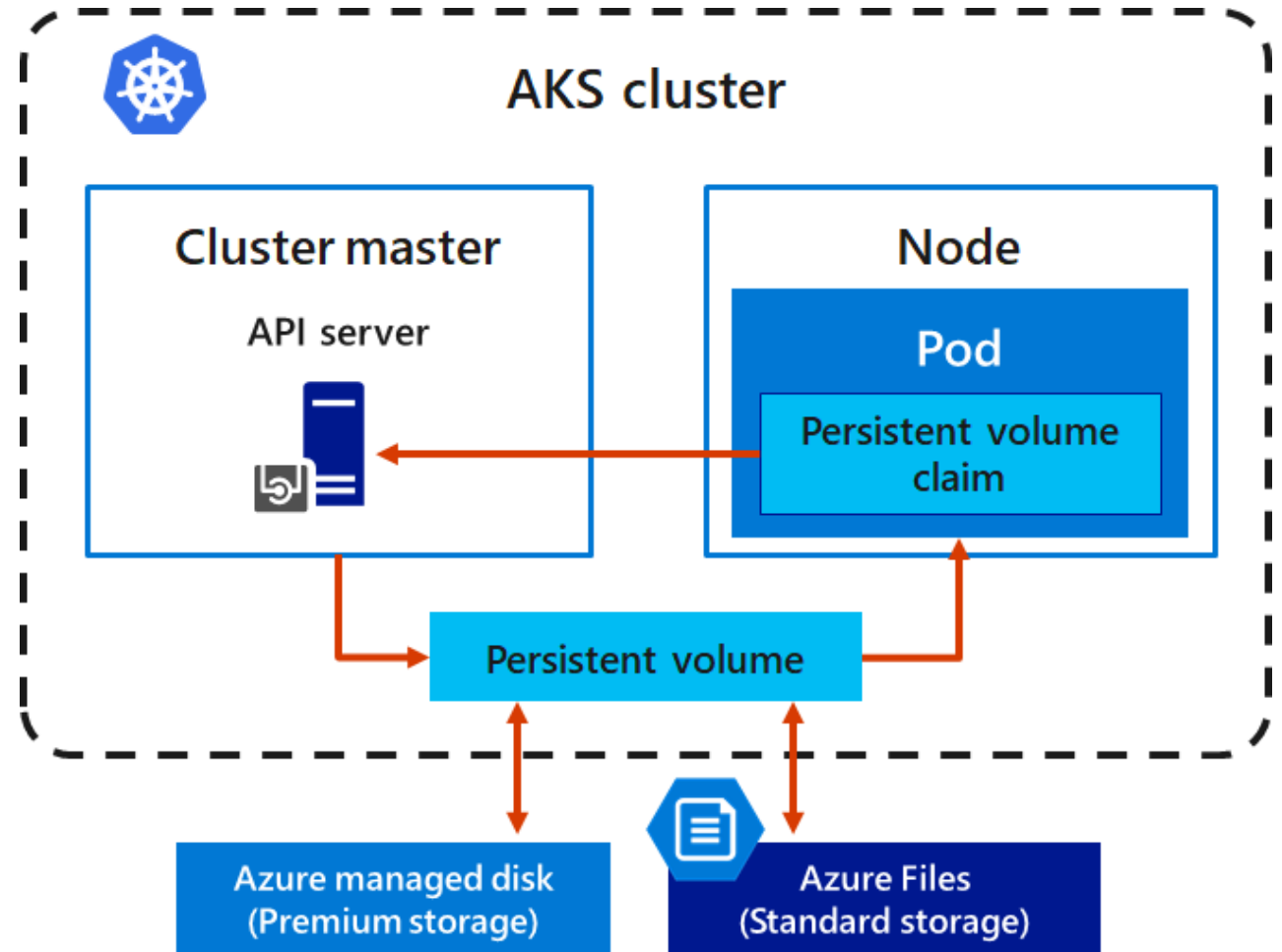
AKS Networking



- Pods run an instance of your application
- Services group pods together to provide network connectivity
- Cluster IP provides internal traffic access
- NodePort provides mapping for incoming direct traffic
- Load balancer has external IP address for incoming non-direct traffic

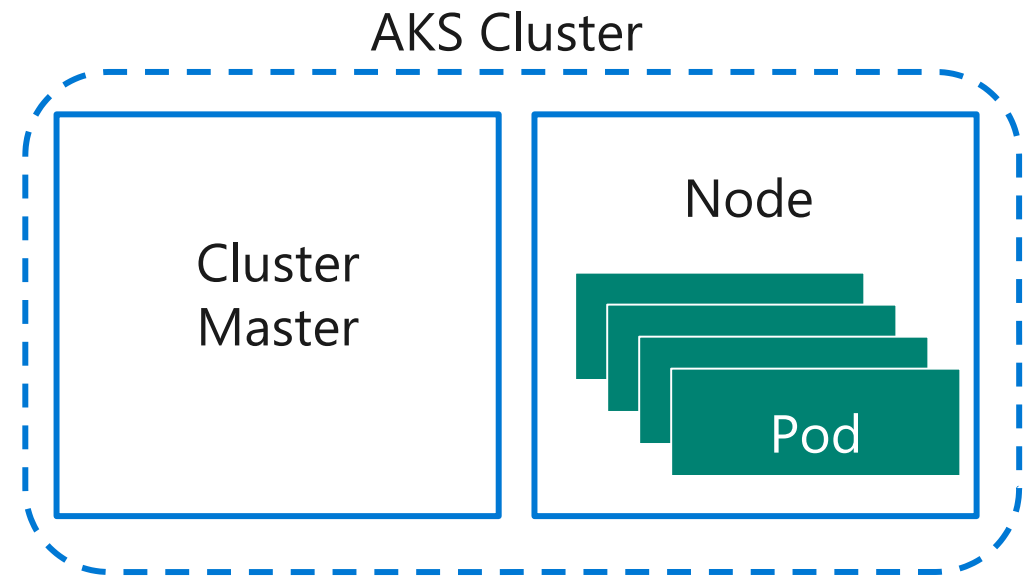
AKS Storage

- Local storage on the node is fast and simple to use
- Local storage might not be available after the pod is deleted
- Multiple pods may share data volumes
- Storage could potentially be reattached to another pod



AKS Security

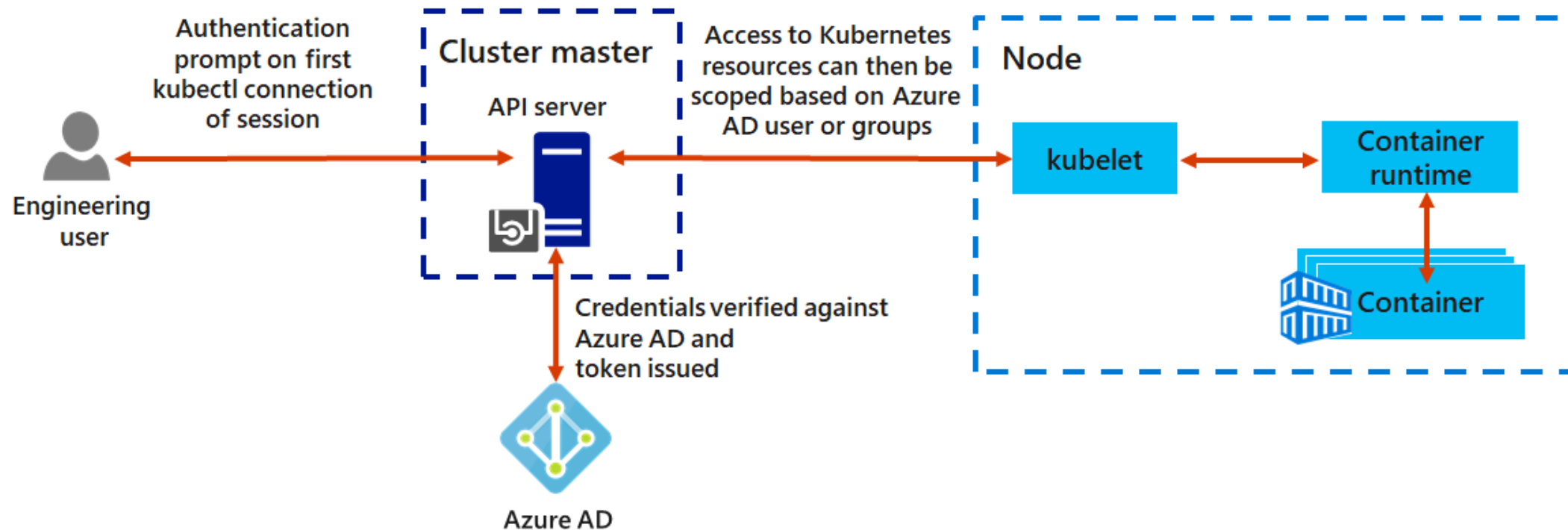
- AKS Cluster – Upgrade orchestration with node cordon and drain
- Cluster Master – fully managed
- Node – automatic OS security patches
- Networks – private virtual networks and network security groups
- Data - Kubernetes secrets for credentials and keys



Consider security for all components

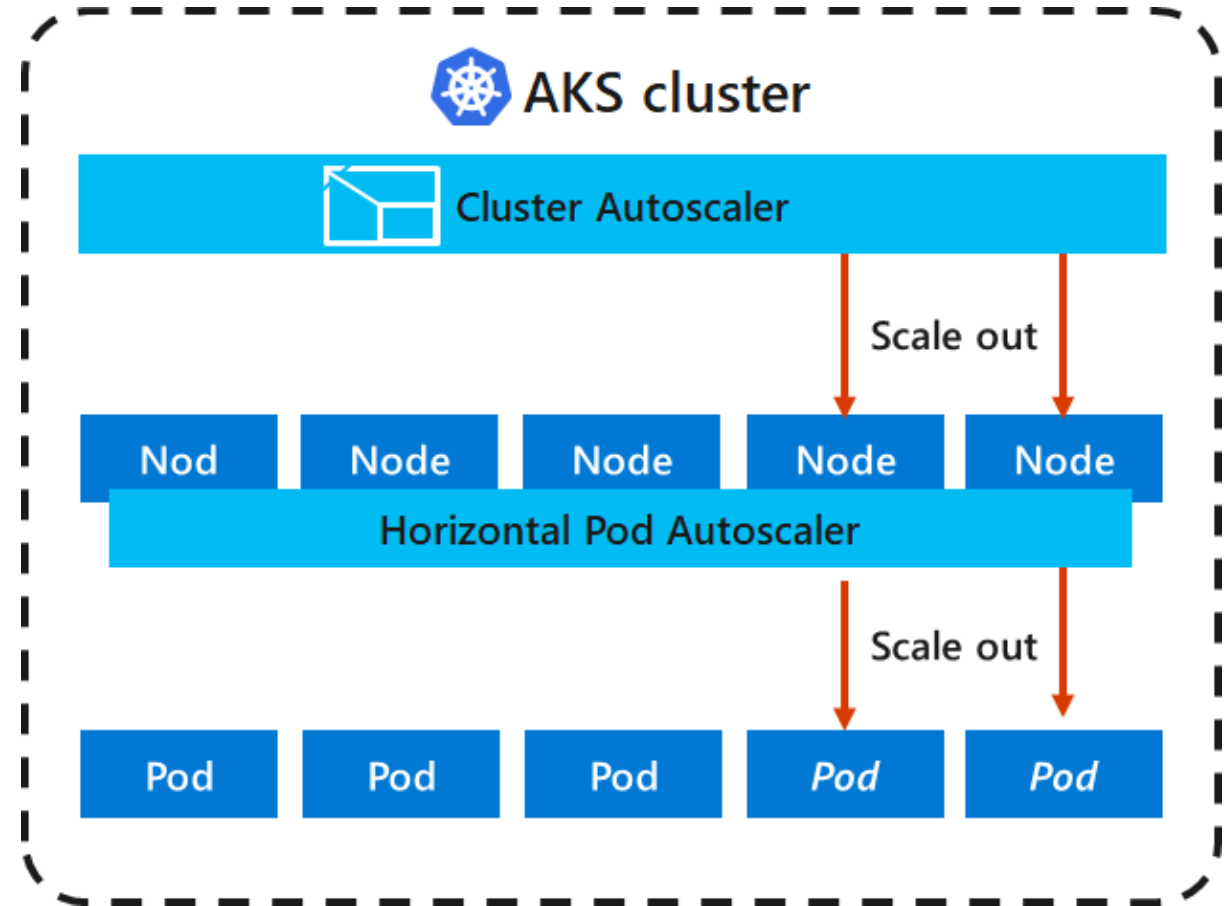
AKS and Azure Active Directory

- Use Azure AD as an integrated identity solution
- Use service accounts, user accounts, and role-based access control



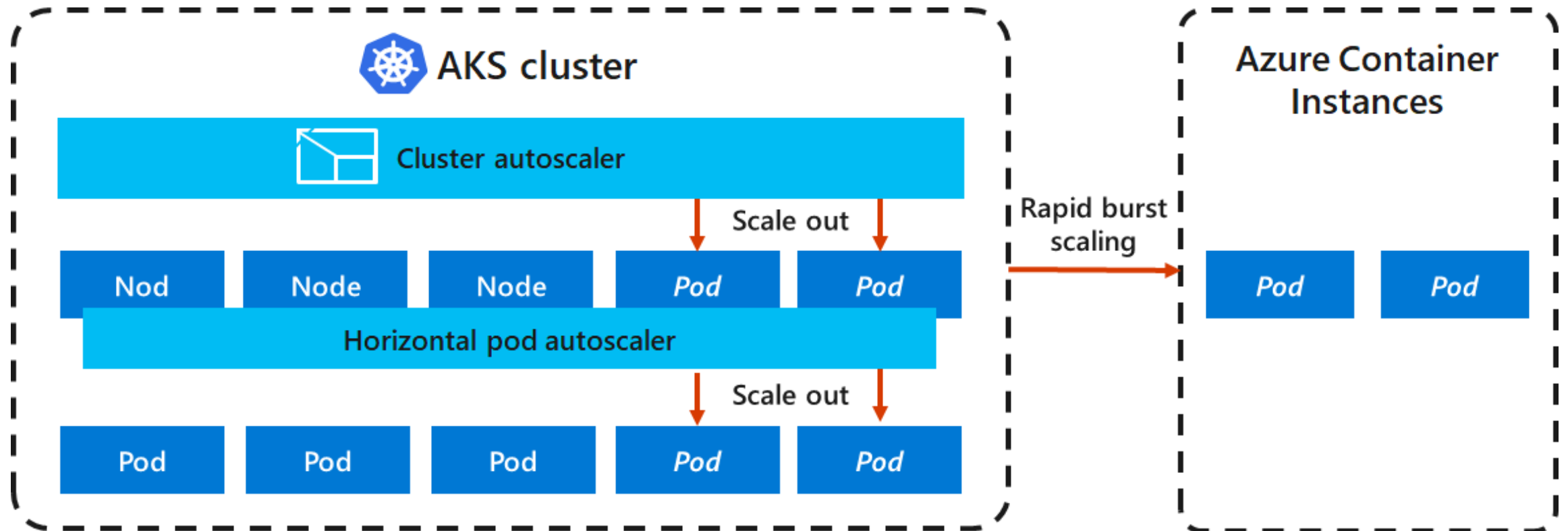
AKS Scaling

- Applications might grow beyond the capacity of a single pod
- Kubernetes has built-in autoscalers
- Cluster autoscaler scales based on compute resources
- Horizontal pod autoscaler scales based on metrics

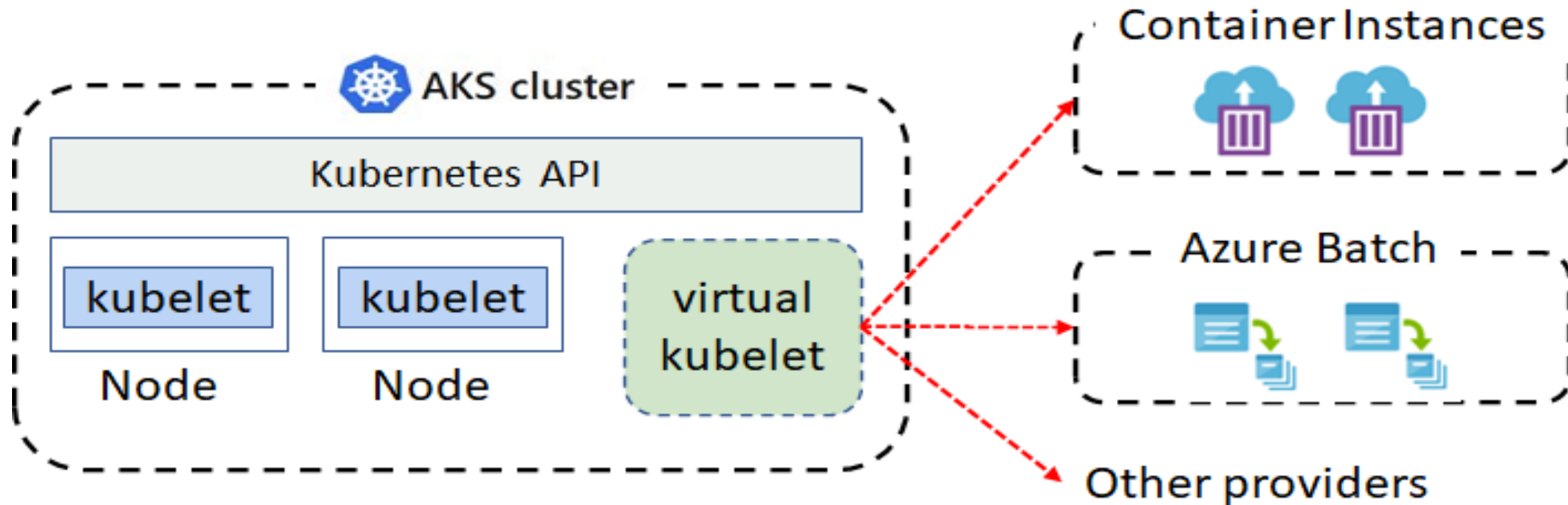


AKS Scaling to ACI

If you need to rapidly grow your AKS cluster, you can create new pods in Azure Container Instances



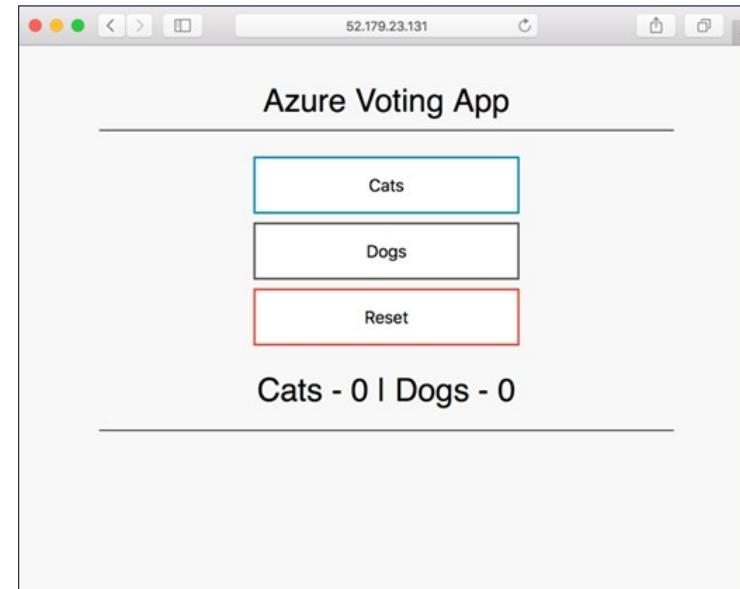
Virtual Kubelet



- Virtual kubelet is an open-source Kubernetes kubelet implementation
- The virtual kubelet registers itself as a node and allows developers to deploy pods and containers with their own APIs
- Supported by an ecosystem of providers

Demonstration - Deploy Azure Kubernetes Service

- Create a Kubernetes service
- Connect to the cluster.
- Test the applications.



Lesson 06: Module Labs and Review



Lab 09a - Implement Web Apps

Lab scenario

You need to evaluate the use of Azure Web apps for hosting Contoso's web sites, hosted currently in the company's on-premises data centers. The web sites are running on Windows servers using PHP runtime stack. You also need to determine how you can implement DevOps practices by leveraging Azure web apps deployment slots.

Objectives

- Task 1: Create an Azure web app
- Task 2: Create a staging deployment slot
- Task 3: Configure web app deployment settings
- Task 4: Deploy code to the staging deployment slot
- Task 5: Swap the staging slots
- Task 6: Configure and test autoscaling of the Azure web app

Lab 09b - Implement Azure Container Instances

Lab scenario

Contoso wants to find a new platform for its virtualized workloads. You identified several container images that can be leveraged to accomplish this objective. Since you want to minimize container management, you plan to evaluate the use of Azure Container Instances for deployment of Docker images.

Objectives

- Task 1: Deploy a Docker image by using the Azure Container Instance
- Task 2: Review the functionality of the Azure Container Instance

Lab 09c - Implement Azure Kubernetes Service

Lab scenario

Contoso has several multi-tier applications that are not suitable to run by using Azure Container Instances. To determine whether they can be run as containerized workloads, you want to evaluate using Kubernetes as the container orchestrator. To minimize management overhead, you want to test Azure Kubernetes Service, including its simplified deployment experience and scaling.

Objectives

- Task 1: Deploy an Azure Kubernetes Service cluster
- Task 2: Deploy pods into the Azure Kubernetes Service cluster
- Task 3: Scale containerized workloads in the Azure Kubernetes service cluster

Module Review

- Module Review Questions
- Microsoft Learn Modules (docs.microsoft.com/Learn)
 - Host a web application with Azure App service
 - Stage a web app deployment for testing and rollback by using App Service deployment slots
 - Scale an App Service web app to efficiently meet demand with App Service scale up and scale out
 - Dynamically meet changing web app performance requirements with autoscale rules
 - Capture and view page load times in your Azure web app with Application Insights
 - Build a containerized web application with Docker
 - Run Docker containers with Azure Container Instances
 - Introduction to the Azure Kubernetes Service